

CLAIMS

I claim:

1. A method of replicating state information among a plurality of collaborating servers connected to a network, the method comprising:

 determining at a subscriber server from information stored on a client computer whether an event has occurred on a publisher server which event implicates a need for state change on the subscriber server; and

 if such an event has occurred, replicating state effects of the event into state on the subscriber server.

2. The method of claim 1, wherein the step of determining at a subscriber server from information stored on a client computer whether an event has occurred on a publisher server further comprises:

 receiving an event queue stored on the client computer to the subscriber server;
 determining whether any events recorded in the event queue are not yet replicated on the subscriber server; and

 determining whether any such events require replication on the subscriber server.

3. The method of claim 2, wherein the step of determining whether any events in the event queue are not yet replicated on the subscriber server further comprises comparing the event queue retrieved from the client computer with a most-recently-experienced event queue recorded by the subscriber server.

4. The method of claim 3, wherein the event queue retrieved from the client computer is recorded as the most-recently-experienced event queue by the subscriber server following the comparison.

5. The method of claim 2, wherein the step of determining whether any unreplicated events in the event queue require replication on the subscriber server further comprises: determining what state on the subscriber server is relevant to the subscriber server at the time; and

determining if replicating such an event would effect change to such state.

6. The method of claim 2, wherein the step of determining whether any unreplicated events in the event queue require replication on the subscriber server further comprises determining if undertaking replication of one such event can be skipped by the subscriber server due to the existence of another such event whose replication by the subscriber server would suffice for the first.

7. The method of claim 1, wherein the step of replicating the state effects of an event into state on the subscriber server further comprises copying state data from the event queue into subscriber server state.

8. The method of claim 1, wherein the step of replicating the state effects of an event into state on the subscriber server further comprises copying state data from another server computer into subscriber server state.

9. The method of claim 8, wherein the state data is copied by the subscriber server over the network from the publisher server.

10. The method of claim 8, wherein the state data is copied by the subscriber server over the network from a database server.

11. The method of claim 8, wherein the state data copied by the subscriber server over the network from another server is converted from a first data format into a second data format.

12. The method of claim 1, further comprising:

processing an event on a publisher server which causes state change on the publisher server;

retrieving an event queue stored on a client computer to the publisher server;

adding a new event descriptor characterizing the event to the event queue; and

sending the event queue from the publisher server to be stored on the client computer.

13. The method of claim 2, wherein the event queue is stored on the client computer within a transport mechanism of a cookie.

14. The method of claim 2, wherein the event queue includes:

one or more event descriptors;

a uniqueness provision; and

a specified duration.

15. The method of claim 12, wherein each event descriptor includes:

a characterization of the general type of event;

zero or more arguments characterizing the specific instance of the event;

a discernible ordinal position within an event queue.

16. The method of claim 14, wherein the uniqueness provision is a unique value in the form of a timestamp.

17. The method of claim 14, wherein the uniqueness provision is a unique value in the form of a pseudorandom datum.

18. The method of claim 12, wherein the step of retrieving an event queue stored on a client computer to a publisher server further comprises allocating an initial event queue if no event queue is yielded by the retrieval.

19. The method of claim 12, wherein the step of adding a new event descriptor characterizing an event to an event queue further comprises adding the event descriptor such that it is ordinally maximal within the event queue.

20. The method of claim 12, wherein the step of adding a new event descriptor characterizing an event to an event queue further comprises removing those event descriptors from the event queue which are no longer needed by any collaborating server.

21. The method of claim 12, wherein the event queue sent to the client computer is recorded as the most-recently-experienced event queue by the publisher server.

22. A system for replicating state information among a plurality of collaborating servers connected to a network, the system comprising:

 logic configured to determine at a subscriber server from information stored on a client computer whether an event has occurred on a publisher server which event implicates a need for state change on the subscriber server; and

 logic configured to replicate state effects of the event into state on the subscriber server, if such an event has occurred.

23. The system of claim 22, wherein the logic configured to determine further comprises:

 logic configured to retrieve an event queue stored on the client computer to the subscriber server;

 logic configured to determine whether any events recorded in the event queue are not yet replicated on the subscriber server; and

 logic configured to determine whether any such events require replication on the subscriber server.

24. The system of claim 23, wherein the logic configured to determine whether any unreplicated events in the event queue require replication on the subscriber server further comprises:

logic configured to determine what state on the subscriber server is relevant to the subscriber server at the time; and

logic configured to determine if replicating such an event would effect change to such state.

25. The system of claim 22, further comprising:

logic configured to process an event on a publisher server which causes state change on the publisher server;

logic configured to retrieve an event queue stored on a client computer to the publisher server;

logic configured to add a new event descriptor characterizing the event to the event queue; and

logic configured to send the event queue from the publisher server to be stored on the client computer.